

## **BAB III**

### **LANDASAN TEORI**

Pada bab ini akan dijelaskan mengenai teori dasar yang digunakan oleh penulis sebagai acuan dalam membangun aplikasi. Karena topik dari penulis memiliki sedikit nilai fotografi, dan penggunaan sistem layanan berbasis lokasi, serta penggunaan *Database management system* untuk berinteraksi dengan database, oleh karena itu maka perlu untuk mengetahui apa arti dari istilah tersebut.

#### **2.1 Pengertian Fotografi**

Fotografi berasal dari kata foto yang berarti cahaya dan grafis yang berarti gambar. Dengan berkembangnya teknologi digital yang sangat pesat saat ini bahkan hampir semua orang.

Secara harfiah fotografi bisa diartikan sebagai teknik melukis dengan cahaya. Fotografi merupakan gabungan ilmu, teknologi, dan seni. Perpaduan yang harmonis antara ketiganya bisa menghasilkan sebuah karya yang mengagumkan. Tentunya dengan skill serta sentuhan seni sang fotografer, sebuah foto bisa menjadi berarti.

Fotografi memiliki bermacam-macam manfaat dan tujuan baik untuk dokumentasi, penelitian, maupun sebagai media dalam ranah estetika (Segara, 2012).

#### **2.2 Sistem Layanan Berbasis Lokasi**

Sistem Layanan Berbasis Lokasi atau sering disebut Location Based Service (LBS) merupakan penggabungan antara proses dari layanan ponsel

dengan geografis dari penggunaannya. Poin pentingnya adalah ketika posisi target, di mana sebuah target bisa jadi adalah pengguna LBS itu sendiri atau entitas lain yang tergabung dalam suatu layanan (Kupper & Treu, 2005).

LBS memberikan layanan personalisasi kepada pengguna ponsel yang disesuaikan dengan lokasi mereka saat ini. LBS membuka pasar baru bagi pengembang, operator jaringan selular, dan penyedia layanan untuk mengembangkan dan memberikan nilai tambah layanan diantaranya untuk memantau kebiasaan mengemudi, mencari karyawan, mencari rute terdekat atau untuk menentukan rute ke lokasi yang ditentukan dari posisi saat ini (Rani, et al., 2012).

Secara garis besar jenis Layanan Berbasis Lokasi juga dapat dibagi menjadi dua, yaitu:

1. *Pull Service*: Layanan diberikan berdasarkan permintaan dari pelanggan akan kebutuhan suatu informasi. Jenis layanan ini dapat dianalogikan seperti mengakses suatu web pada jaringan internet.
2. *Push Service*: Layanan ini diberikan langsung oleh service provider tanpa menunggu permintaan dari pelanggan, tentu saja informasi yang diberikan tetap berkaitan dengan kebutuhan pelanggan.

Salah satu perangkat mobile yang mendukung sistem layanan berbasis lokasi ialah android.

### 3.3 *Android*

*Android* adalah sistem operasi untuk perangkat mobile yang mana terdiri dari sebuah sistem operasi, dan aplikasi utama yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Aplikasi *android* ditulis dalam bahasa pemrograman *Java* dengan *Java Libraries* yang dikembangkan Google.

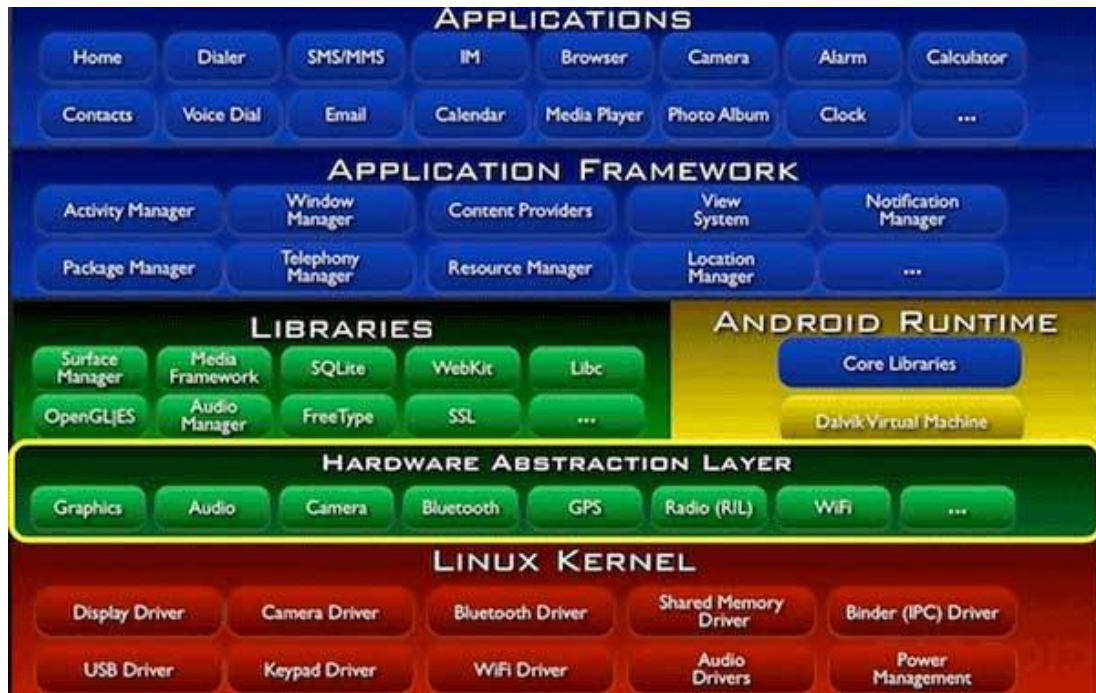
*Android* bukan sekedar hanya untuk perangkat mobile saja, *android* merupakan sebuah sistem operasi yang dikemas sedemikian rupa sehingga dapat digunakan untuk berbagai perangkat yang menggunakan layar (Simmonds, 2010).

Aplikasi berbasis *android* dapat dengan mudah dibuat dalam IDE (*Integrated Development Environment*) Eclipse dengan bantuan eksistensi *android* yang disebut ADT (*Android Development Tools*) yang fungsinya diantara lain untuk:

1. Memungkinkan akses ke *Android Development Tools* lainnya.
2. Memungkinkan akses *New Project Wizard* untuk aplikasi *android*.
3. Mengotomasi dan menyederhanakan proses pembangunan aplikasi *android*.
4. Menyediakan *Android Code Editor* untuk membantu pengguna menulis file XML dengan

valid untuk Android Manifest dan file resource lainnya.

5. Mempublikasikan aplikasi menjadi signed APK agar dapat didistribusikan kepada pengguna lain. (Android Developer, 2010).



Gambar 3. 1. Arsitektur Android

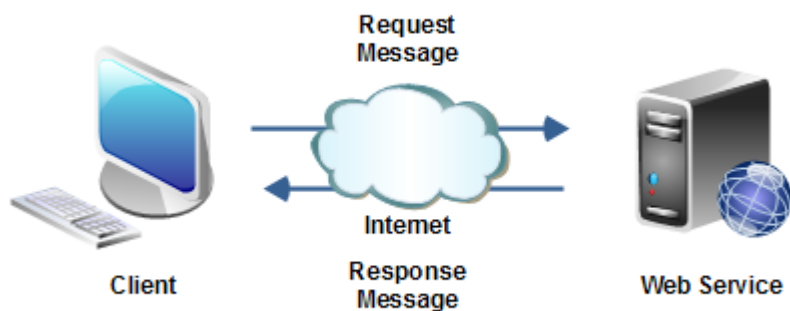
Pada gambar 3.1 menjelaskan bahwa android mempunyai kebutuhan cukup untuk penulis mengembangkan aplikasi. Dan dengan memanfaatkan web service penulis dapat lebih mudah dalam melakukan pertukaran data.

#### 3.4 Web Service

Web Service adalah sebuah teknologi yang bisa digunakan untuk membuat sebuah aplikasi yang bersifat *platform-independent*. Web Service dapat menghubungkan antar web yang berbeda, aplikasi yang berbeda, bahkan sampai device yang berbeda

dalam sistem operasi yang berbeda pula (Hendrawan, 2006).

*Web Service* dapat menjalankan operasi-operasi termasuk akses data, memperbarui database, mengatur integritas data, mengirim respon dari permintaan yang dikirim oleh pengguna. *Web service* tidak memiliki tampilan karena *web service* termasuk dalam *Business-Service tier*. Artinya di dalam *web service* hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh aplikasi lainnya. Pada penggunaan pada sistem perancangan ini digunakan *web service* yang berupa file ber-ekstensi *.php* untuk dapat menjembatani antara aplikasi *mobile phone* dengan basis data. *Web service* diletakkan tempatnya sama dengan lokasi basis data. Penggunaannya dengan menggunakan *HTTP Access* dimana hasil balikan dari *web service* adalah data dengan ekstensi *.json*.



*Gambar 3. 2 Arsitektur Web Service*

Salah satu pemanfaatan *web service* yang dibutuhkan untuk mengembangkan aplikasi penulis ialah *Google Maps API* dan *Google Cloud Massaging*. Serta penggunaan bahasa dalam membentuk *web*



service, penulis menggunakan bahasa PHP, serta menggunakan format pertukaran data menggunakan JSON.

### 3.5 *PHP*

PHP adalah bahasa *scripting language* yang dirancang secara khusus untuk penggunaan pada web. PHP kependekan dari *HyperText Preprocessor*. Seperti bahasa pemrograman web lainnya PHP memproses seluruh perintah yang berada dalam skrip

PHP didalam web browser dan menampilkan output-nya ke dalam web browser klien. PHP adalah bahasa *scripting* yang menghasilkan *output* HTML ataupun *output* lain sesuai keinginan program yang dijalankan pada *server side* (Suryatiningsih, 2010).

### 3.6 *Google Maps API*

*Google Maps API* merupakan pengembangan teknologi dari google yang digunakan untuk menanamkan google Map di suatu aplikasi yang tidak dibuat oleh Google. *Google Maps API* adalah suatu library yang berbentuk javascript yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan (Djanali, et al., 2012).

Dalam perkembangannya *Google Maps API* diberikan kemampuan untuk mengambil gambar peta statis. Melakukan *geocoding*, dan memberikan penutun arah. *Google Maps API* bersifat gratis

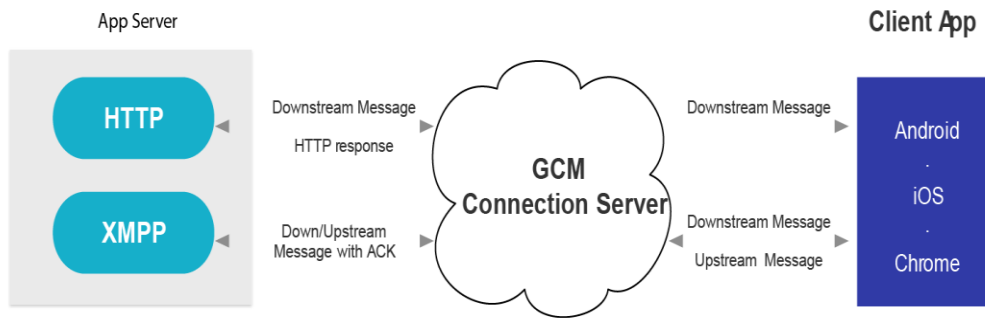
untuk publik. Penggunaan *Google Maps API* pada pengembangan aplikasi android dengan menggunakan Eclipse dan komputer menggunakan sistem operasi windows.

*Google Maps Api* memiliki kekurangan dan kelebihan. Kekurangan yang dimiliki oleh *Google Maps Api* adalah jika ingin melakukan akses harus terdapat layanan internet pada perangkat yang digunakan. Sedangkan kelebihannya adalah adanya dukungan penuh yang dilakukan Google sehingga terjamin dan bervariasi fitur yang ada pada *Google Maps API*. Banyaknya pengembang yang menggunakan *Google Maps API* telah memudahkan dalam mencari referensi dalam pengembangan aplikasi.

### 3.7 *Google Cloud Messaging*

Merupakan layanan gratis yang disediakan oleh Google yang memungkinkan para pengembang aplikasi untuk bertukar pesan antara server dan aplikasi di klien. GCM dapat mengirimkan data maksimal sebesar 4kb ke klien. GCM menangani semua aspek mulai dari data dikirim hingga data diterima oleh klien.

Data dikirim menggunakan protokol *HTTP* atau *XMPP* melalui *GCM Connection Server* yang disediakan oleh Google. Gambar 3.3 merupakan arsitektur GCM yang dibuat oleh Google. (Google, 2016)



*Gambar 3. 3 Arsitektur GCM*

Berikut ini adalah penjelasan dari gambar 3.3.

- Google Connection Server* yang disediakan oleh google akan menerima pesan yang dikirimkan oleh server aplikasi milik kita dan mengirimkan pesan tersebut ke aplikasi di klien.
- App Server* merupakan server yang harus disediakan oleh kita sendiri serta harus menggunakan protokol HTTP ataupun XMPP untuk berinteraksi dengan *GCM Connection Server*.
- Client App* merupakan aplikasi di klien yang akan menerima pesan yang dikirimkan melalui *Google Connection Server* dan untuk mengirim atau menerima pesan dari GCM maka aplikasi ini harus didaftarkan terlebih dahulu ke GCM dan akan mendapatkan kode identifikasi yang bersifat unik atau disebut juga *token*.

### 3.8 JSON dan XML

*JSON (JavaScript Object Notation)* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer.



Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 -

Desember

```
{
    "id" :    "2",
    "name" :   "Table of
Contents"
}
```

1999.

*Gambar 3. 4 Simple encoding JSON*

Sedangkan XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet.

```
<data>
  <id>1</id>
  <name>PHP
</name>
```

*Gambar 3. 5 Simple encoding XML*

Perbandingan antara JSON dan XML dapat diuraikan seperti :

JSON :

1. Mendukung array
2. Mendukung pembuatan Object
3. Sintak pendek
4. Dapat berpadu dengan AJAX
5. Akses data cepat

XML :

1. Tidak mendukung array
2. Tag dibuat manual/dideklarasikan oleh programmer
3. Ukuran data besar
4. Harus menggunakan XML DOM jika ingin memetakan teks/data
5. Dapat berpadu dengan AJAX

Membandingkan perbedaan antara dua jenis format pertukaran data saat ini. Hasil penelitian menunjukkan bahwa JSON lebih cepat dan menggunakan sumber daya yang lebih sedikit daripada XML. JSON dan XML menyediakan kelebihan yang unik, tetapi pentingnya kinerja dan pemanfaatan sumber daya harus dipahami ketika membuat keputusan antara format pertukaran data. (Nurseitov, et al., 2009 )

### 3.9 Database Management System

Database Management System berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. DBMS terdiri dari set program pengelola untuk menambahkan data, menghapus data, mengambil data, dan membaca data. Database adalah kumpulan file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu database menunjukan satu kumpulan data yang di pakai dalam satu lingkup perusahaan, instansi (Kristanto, 2004).